ELSEVIER International Journal of Approximate Reasoning 25 (2000) 187–215

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

www.elsevier.com/locate/ijar

Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing ☆

Oscar Cordón [a], Francisco Herrera [b,*], Pedro Villar [b]

[a] *Department of Computer Science and Artificial Intelligence, University of Granada, ETS de Ingeniera Informatica, Avda. Andalucia 38, 18071 Granada, Spain*

[b] *Department of Computer Languages and Systems, University of Vigo, 32004 Ourense, Spain*

Received 1 November 1999; accepted 1 May 2000

Abstract

In this contribution, we will analyse the importance of the fuzzy partition granularity for the linguistic variables in the design of fuzzy rule-based systems (FRBSs). In order to put this into effect, we will study the FRBS behaviour considering uniform fuzzy partitions with the same number of labels for all the linguistic variables, and considering uniform fuzzy partitions with any number of labels for each linguistic variable. We will present a method based on Simulated Annealing (SA) in order to obtain a good uniform fuzzy partition granularity that improves the FRBS behaviour. It is an efficient granularity search method for finding a good number of labels per variable. © 2000 Elsevier Science Inc. All rights reserved.

Keywords: Fuzzy rule-based systems; Data base; Fuzzy partition; Granularity; Simulated annealing

☆ This research has been supported by CICYT projects TIC96-0778 and PB98-1319.

* Corresponding author. Tel.: +34-58-24-40-19; fax: +34-58-24-33-17.

E-mail addresses: ocordon@decsai.ugr.es (O. Cordón), herrera@decsai.ugr.es (F. Herrera), pvillar@uvigo.es (P. Villar).

PII: S0888-613X(00)00052-9

1. Introduction

Fuzzy rule-based systems (FRBSs) represent one of the most important areas for the application of fuzzy set theory. These systems constitute an extension of classical rule-based systems, because they deal with fuzzy rules instead of classical logic rules. They have been successfully applied to a wide range of problems presenting uncertainty and vagueness in different ways [2,15,24].

An FRBS presents two main components: (1) the inference system, which implements the fuzzy inference process needed to obtain an output from the FRBS when an input is specified, and (2) the knowledge base (KB), which represents the knowledge about the problem being solved. The KB is composed by the rule base (RB) containing the collection of fuzzy rules, and by the data base (DB) containing the membership functions of the fuzzy partitions associated to the linguistic variables.

Two main tasks need to be performed to design an FRBS for a specific problem: to select the fuzzy operators involved in the inference system, i.e., to define the way in which the fuzzy inference process will be performed, and to derive an appropriate KB about the problem under solving. The accuracy of the FRBS in the solving of this problem will depend directly on both components.

Focusing on the second design task, many approaches have been presented to automatically learn the RB from numerical information (input–output data pairs representing the system behaviour) when there is no knowledge provided by an human expert. However, there is not a similar effort for deriving the DB, although its design is a critical task since most of the RB learning methods assume the existence of a previously defined DB, and thus it will significantly condition the behaviour of the final FRBS.

A very common way to proceed involves considering uniform fuzzy partitions with the same number of terms for all the linguistic variables of the problem, that is, the same granularity. The aim of this article is to analyse the influence of the granularity of the fuzzy partitions in the FRBS performance. To be precise, we will deal with this problem from a double perspective:

- We will try to give an answer to the question: *is it a good operation mode to consider uniform fuzzy partitions with the same number of labels for all the linguistic variables*?
- We will also develop an efficient method for obtaining good uniform fuzzy partitions finding a good granularity per linguistic variable.

To do so, we will work with different RB automatic learning methods and we will compare their behaviour when considering DBs with a different number of linguistic terms for each linguistic variable. The membership functions considered will always be triangular-shaped, symmetrical and uniformly distributed, thus making the granularity of the fuzzy partitions the unique parameter of the DB having influence on the learning method and, consequently, on the final FRBS behaviour. Moreover, we propose simulated annealing (SA) as the method to search for a good uniform fuzzy partition granularity, i.e., a granularity that produces an FRBS with good accuracy, and in some cases, the one with the best behaviour.

This paper is organized as follows. In Section 2, we present the role of the DB in the FRBS design process. In Section 3, we analyse the influence of the uniform fuzzy partition granularity on the FRBS behaviour taking three real-world applications as a base. First, we study the FRBS behaviour considering the same number of labels in each linguistic variable (Section 3.1), and later, considering any number of labels in each linguistic variable (Section 3.2). Finally, the conclusions of the study are presented. In Section 4, we present a SA method for obtaining a uniform fuzzy partition granularity with good behaviour and validate it on the said problems. In Section 5, some concluding remarks are provided. A short description of the RB learning methods used in the paper is given in Appendix A, while the characteristics of the problems considered as benchmarks can be found in Appendix B. Finally, the SA procedure is briefly described in Appendix C.

2. The role of the data base in the design of FRBSs

The composition of the KB of an FRBS directly depends on the problem being solved. The best situation is when there is a human expert able to express his/her knowledge in the form of fuzzy rules, thus providing the definitions for the DB (the relevant input and output linguistic variables for the system, the term sets for all of them and the membership functions of the fuzzy sets defining their meaning) and for the RB (the fuzzy rules themselves). Unfortunately, this situation is not very common: usually the expert is not able to provide all this information or there is no expert information about the problem under solving.

In the last few years, many approaches have been proposed to solve this problem. These approaches try to automatically learn the RB from numerical information (input–output data pairs representing the system behaviour), using different techniques such as ad hoc data-driven algorithms [2,4,18,31], least square methods [2], gradient descent algorithms [22], hybrid methods between the latter two ones [20], clustering algorithms [32], neural networks [29] and genetic algorithms (GAs) [7].

As we have mentioned, there is not a similar effort for deriving the DB. However, the DB has a significant influence on the FRBS performance. In fact, studies such as the ones developed in [3,33] show, for the case of Fuzzy PI controllers, that the system performance is much more sensitive to the choice of the semantics in the DB than to the composition of the RB. Considering a previously defined RB, the performance of the Fuzzy controller is sensitive to four aspects: scaling factors, peak values, width values and rules. For this reason, some approaches try to improve the preliminary DB definition considered once the RB has been derived. To do so, a tuning process considering the whole KB obtained (the preliminary DB and the derived RB) is used a posteriori to adjust the membership function parameters to improve the FRBS behaviour (for some examples of these kinds of methods, based on neural networks and GAs, refer to [3,6,16,20]). Nevertheless, the tuning process only adjusts the shapes of the membership functions and not the number of linguistic terms in each fuzzy partition, which remains fixed from the beginning of the design process. Other more sophisticated approaches to learn the different DB components can be found in [11,12,14,19,26,30].

Usually, the most very common way to proceed for learning the RB considers, as starting point, a DB composed of uniform fuzzy partitions with the same number of terms (usually an odd number between three and seven) for all the linguistic variables existing in the problem. Triangular or trapezoidal-shaped membership functions are usually considered due to their simplicity.

At first sight, the selection of the granularity level in the input and output variable fuzzy partitions does not seem to be a DB design task as important as the choice of the membership function shapes for the linguistic terms. However, the granularity selection plays an important role in many characteristics of the FRBS, such as the accuracy in fuzzy modeling or the smoothness in fuzzy control. Moreover, the granularity of the input variables specifies the maximum number of fuzzy rules that may compose the RB, thus having a strong influence on aspects such as the complexity of the rule learning, the interpretability of the FRBS obtained or its accuracy.

3. Study of the influence of the uniform fuzzy partition granularity on the FRBS behaviour

Typically, the DB is defined by choosing an equal number of linguistic terms for all the variables and by considering uniform fuzzy partitions in the variable universe of discourse for these labels. This choice is not guided by any specific characteristic of the problem, nor by any general rule.

In this section, we analyse the use of three learning methods to explore the problem of granularity selection. [1] First, we constrain all the variables to have the same number of labels. Later, each variable is allowed to have any number

[1] See Appendix A for a description of them: Wang and Mendel [31], Cordón and Herrera [10] and Descriptive–Mogul [6] learning methods.

of labels. In both cases, we used the interval {3–9} as possible values for the number of linguistic terms.

To compare the behaviour of the different FRBSs obtained, we consider three real-world applications: *Low voltage line length problem*, *Optimal electrical network problem* and *Rice taste evaluation problem*. The description of the benchmark problems can be found in Appendix B. The set of data pairs of every benchmark considered has been divided into two subsets, denoted *training set* and *test set*. The former is used by the learning methods to derive the RB composition, while the latter is used to evaluate the prediction ability of the generated fuzzy models.

The mean square error of the FRBS over the training and test sets ($\mathrm{MSE_{tra}}$ and $\mathrm{MSE_{tst}}$) is used as a comparison measure for the different FRBSs obtained

$$\frac{1}{2|E|}\sum_{e_l\in E}(ey^l - S(ex^l))^2$$

with E being the example set (training or test), $S(ex^l)$ being the output value obtained from the FRBS when the input variable values are $ex^l = (ex_1^l, \ldots, ex_n^l)$, and ey^l being the known desired value.

3.1. FRBSs with the same number of labels for each variable

In this part of the study, the three learning methods were run with the same number of labels for all the variables. Each method was run seven times for each benchmark. The results, the $\mathrm{MSE_{tra}}$, the $\mathrm{MSE_{tst}}$ and the number of rules (#R), are shown in Tables 1–3 (where the best $\mathrm{MSE_{tst}}$ value found in each method appears in bold type).

The analysis of these results leads us to the following conclusions:

- Different learning methods generate the best FRBS design using a different value for the fuzzy partition granularity.
- The difference in the FRBS accuracy is significant enough to validate the importance of granularity selection as an important task that must be adequately analysed during the RB learning process.

The $\mathrm{MSE_{tst}}$ obtained in Tables 1–3 are also showed in Fig. 1. In this graphic, it can be seen that the general behaviour of the three learning methods is similar for each problem, but the best results are obtained using a different number of linguistic terms.

On the other hand, it is interesting to observe that an excessively high number of labels can cause an over-fitting problem. Particularly, considering the WM and D-Mogul methods in the *low voltage line length* problem (Table 1), the FRBSs with best $\mathrm{MSE_{tra}}$ use nine labels, while the value of the $\mathrm{MSE_{tst}}$ in both cases is significantly worse than the one obtained by the FRBSs with six labels (best $\mathrm{MSE_{tst}}$). The over-fitting problem is apparent when we increase the number of labels per variable in the rice taste evaluation problem

Table 1
Results for the low voltage line length problem

		WM		CH		D-Mogul	
		MSE	#*R*	MSE	#*R*	MSE	#*R*
3	MSE_{tra}	594276.31	7	322227.62	9	186172.75	12
Lab.	MSE_{tst}	626566.81		293986.96		162589.45	
4	MSE_{tra}	301732.00	10	292714.53	14	200628.48	16
Lab.	MSE_{tst}	270747.46		270349.84		180553.01	
5	MSE_{tra}	298446.03	13	329726.25	20	166484.81	20
Lab.	MSE_{tst}	282058.15		306325.78		170550.12	
6	MSE_{tra}	239563.01	18	317516.65	27	161810.56	31
Lab.	MSE_{tst}	**194842.84**		311065.81		**157403.32**	
7	MSE_{tra}	222622.70	24	267923.96	32	167621.18	35
Lab.	MSE_{tst}	240018.25		249523.87		207597.64	
8	MSE_{tra}	241716.73	28	199421.39	42	149415.43	61
Lab.	MSE_{tst}	216651.60		**180000.48**		168025.17	
9	MSE_{tra}	197613.43	29	201272.89	47	148068.64	72
Lab.	MSE_{tst}	283645.56		224805.70		205396.95	

Table 2
Results for the optimal electrical network problem

		WM		CH		D-Mogul	
		MSE	#*R*	MSE	#*R*	MSE	#*R*
3	MSE_{tra}	197312.81	28	419500.09	64	106260.83	17
Lab.	MSE_{tst}	174399.79		325602.87		99481.82	
4	MSE_{tra}	121435.94	49	139026.67	139	81527.64	36
Lab.	MSE_{tst}	84012.14		126107.35		74217.22	
5	MSE_{tra}	71294.41	66	127106.94	268	57071.82	53
Lab.	MSE_{tst}	80933.96		148750.57		59060.91	
6	MSE_{tra}	47972.61	92	87471.60	402	34605.73	84
Lab.	MSE_{tst}	50987.58		94282.78		44286.36	
7	MSE_{tra}	57352.08	104	72563.13	554	42223.66	125
Lab.	MSE_{tst}	49075.74		79111.33		49247.49	
8	MSE_{tra}	36906.85	130	42735.82	543	26690.70	197
Lab.	MSE_{tst}	43330.26		**53596.96**		**32783.08**	
9	MSE_{tra}	32337.47	130	50068.17	904	26850.45	200
Lab.	MSE_{tst}	**33504.99**		55617.92		33752.10	

Table 3
Results for the rice taste evaluation problem

		WM		CH		D-Mogul	
		MSE	#*R*	MSE	#*R*	MSE	#*R*
3	MSE_{tra}	0.003372	20	0.003772	180	0.002429	8
Lab.	MSE_{tst}	0.002801		0.006090		0.003229	
4	MSE_{tra}	0.002618	27	0.001934	357	0.002609	21
Lab.	MSE_{tst}	**0.002663**		0.001731		**0.002064**	
5	MSE_{tra}	0.001796	44	0.001121	482	0.001160	40
Lab.	MSE_{tst}	0.010306		0.001624		0.011107	
6	MSE_{tra}	0.001741	56	0.000998	680	0.001248	59
Lab.	MSE_{tst}	0.012189		**0.000917**		0.018189	
7	MSE_{tra}	0.000801	59	0.000757	846	0.000803	63
Lab.	MSE_{tst}	0.017895		0.001454		0.018371	
8	MSE_{tra}	0.000773	68	0.000607	1085	0.000588	91
Lab.	MSE_{tst}	0.029282		0.010365		0.029843	
9	MSE_{tra}	0.000524	70	0.000338	1272	0.000353	115
Lab.	MSE_{tst}	0.034039		0.020294		0.045417	

(Table 3). The reason is the small set of available data pairs. It is known that for a more complex model structure, a larger training data set must be used to obtain a well-performing model. In our case, it is easier to learn the behaviour of the examples contained in the training set by increasing the number of labels. However, the generalization capability is lost in the FRBS obtained. Hence, higher granularity levels cause smaller MSE_{tra} and larger MSE_{tst}.

In view of these conclusions, we suggest to run the learning method as many times as possible values for the number of labels considered, maintaining this value equal for all the variables. By following this approach, we were able to find the FRBS with best accuracy with seven runs (3–9 labels). The cost of this process is relatively low, although it should be considered that some kinds of methods have a run time that grows exponentially with the number of labels.

3.2. FRBSs with any number of labels for each variable

Next, we will analyse the behaviour of the FRBSs obtained when considering different number of labels for each individual variable. The study has been performed only with the ad hoc data-driven methods (WM and CH), because we want to find the best granularity (according to the MSE_{tra} or MSE_{tst}) using deterministic methods. Carrying out this study with non-deterministic methods that can give a different FRBS definition for different runs (such as D-Mogul) is more complicated. This would require a large number of runs (using different

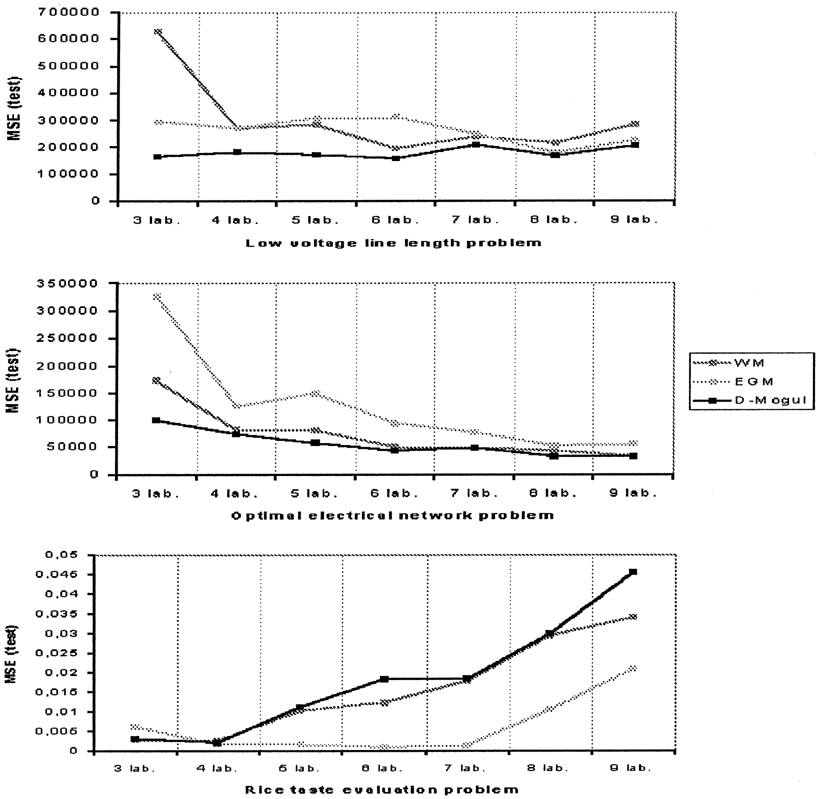

Fig. 1. Comparative of the MSE_{tst} obtained changing the number of labels.

seed values) followed by a statistical analysis of the results. Moreover, non-deterministic methods require a lot of run time in most cases. In our case, considering the said interval (3–9 labels), it would need 7^n runs, n being the number of problem variables and 7 the number of possible values of labels.

The best results obtained in the said study are shown in Table 4.

We can see that the fuzzy partition granularity that shows the best results (both MSE_{tst} and MSE_{tra}) is different in the two methods for all the benchmarks. The reason is that each method uses the information contained in the DB in a different way during the learning process.

At this point, it seems interesting to use the best fuzzy partition granularity for one learning method in the other learning methods, in order to check if these granularity levels produce FRBSs with good performance or, at least, with better behaviour than the best one obtained when considering the same number of labels for all the variables. For this comparison, we will use the fuzzy partition granularity that produces the FRBS with the best MSE_{tst}.

Table 4
Best results with any number of labels

			WM	CH
Low voltage line length problem	Best result in MSE_{tra}	Granularity	**6 9 9**	**8 8 6**
		MSE_{tra}	186904.3	192498.2
		MSE_{tst}	264896.5	167731.5
	Best result in MSE_{tst}	Granularity	**9 6 9**	**7 6 7**
		MSE_{tra}	202370.9	210983.0
		MSE_{tst}	146355.0	152412.4
Optimal electrical network problem	Best result in MSE_{tst}	Granularity	**5 7 7 7 9**	**5 6 9 9 7**
		MSE_{tra}	24867.7	27698.0
		MSE_{tst}	26964.1	26134.3
	Best result in MSE_{tst}	Granularity	**3 4 9 8 8**	**3 6 9 8 7**
		MSE_{tra}	26440.3	27776.1
		MSE_{tst}	24310.9	25914.4
Rice taste evaluation problem	Best result in MSE_{tra}	Granularity	**9 9 9 8 7 9**	**9 9 9 9 7 9**
		MSE_{tra}	0.00042	0.00032
		MSE_{tst}	0.03771	0.01084
	Best result in MSE_{tst}	Granularity	**3 4 8 7 4 5**	**3 5 7 6 3 5**
		MSE_{tra}	0.00159	0.00148
		MSE_{tst}	0.00058	0.00058

Table 5
Results with the best MSE_{tst} (low length voltage line problem)

	Best granularity (same #L)		With best WM granularity (9 6 9)		With best CH granularity (7 6 7)	
	#L	MSE_{tst}	MSE_{tst}	% Improvement	MSE_{tst}	% Improvement
WM	6	194842.8	146355.0	24.8%	154428.4	20.7%
CH	8	180000.4	269079.1	−33.1%	152412.4	15.3%
D-Mogul	6	157403.3	173169.1	−9.1%	167534.3	−6.0%

The results are shown in Tables 5–7 where the first column ("Best Granularity ...") contains the results of the FRBS with best MSE_{tst} obtained for the method considering the same number of labels for every variable. The remaining two columns ("With best ...") show the parameters associated to the FRBS obtained using the best granularity found in the WM and the CH method, respectively. The latter two columns comprise two subcolumns: the MSE_{tst} obtained using the associated number of labels and the percentage improvement of this measure with respect to the MSE_{tst} obtained in the first column.

Table 6
Results with the best MSE_{tst} (optimal electrical network problem)

	Best granularity (same #L)		With best WM granularity (3 4 9 8 8)		With best CH granularity (3 6 9 8 7)	
	#L	MSE_{tst}	MSE_{tst}	% Improvement	MSE_{tst}	% Improvement
WM	9	33504.9	26440.3	21.0%	32413.3	3.2%
CH	8	53596.9	46097.5	13.9%	25914.4	51.6%
D-Mogul	8	32783.0	16482.2	49.7%	24690.0	24.6%

Table 7
Results with the best MSE_{tst} (rice taste evaluation problem)

	Best granularity (same #L)		With best WM granularity (3 4 8 7 4 5)		With best CH granularity (3 5 7 6 3 5)	
	#L	MSE_{tst}	MSE_{tst}	% Improvement	MSE_{tst}	% Improvement
WM	4	0.00266	0.00058	78.1%	0.01002	−276.6%
CH	6	0.00091	0.00150	−64.8%	0.00058	36.2%
D-Mogul	4	0.00206	0.00239	−16.0%	0.01040	−404.8%

As it can be observed, the use of the granularity that produces the best MSE_{tst} for a specific learning method does not always cause good behaviour in another one. In some cases there is a high performance improvement with respect to the best MSE_{tst} found considering the same number of labels for all the variables. In a few cases, the improvement is very small, and in other cases the accuracy decreases.

Now, we present a similar study, but considering the fuzzy partition granularity that produces the FRBS with the best MSE_{tra} (Tables 8–10). The results obtained using the best MSE_{tra} are very similar to the ones obtained using the best MSE_{tst} and the same conclusions can be drawn. The use of the best fuzzy partition granularity in MSE_{tra} for a learning method in another learning method does not always produce good behaviour in the latter one.

Table 8
Results with the best MSE_{tra} (low voltage line length problem)

	Best granularity (same #L)		With best WM granularity (6 9 9)		With best CH granularity (8 8 6)	
	#L	MSE_{tra}	MSE_{tra}	% Improvement	MSE_{tra}	% Improvement
WM	9	197613.4	186904.3	5.4%	216140.7	−9.3%
CH	8	199421.3	195989.7	1.7%	192498.2	3.4%
D-Mogul	9	148068.6	147889.9	0.1%	155151.3	−4.7%

Table 9
Results with the best MSE_{tra} (optimal electrical network problem)

	Best granularity (same #L)		With best WM granularity (5 7 7 7 9)		With best CH granularity (5 6 9 9 7)	
	#L	MSE_{tra}	MSE_{tra}	% Improvement	MSE_{tra}	% Improvement
WM	9	32337.4	24867.7	23.0%	38910.2	−20.3%
CH	8	42735.8	40308.6	5.6%	27698.0	35.1%
D-Mogul	8	26690.7	26805.5	−0.4%	24114.9	9.6%

Table 10
Results with the best MSE_{tra} (rice taste evaluation problem)

	Best granularity (same #L)		With best WM granularity (9 9 9 8 7 9)		With best CH granularity (9 9 9 9 7 9)	
	#L	MSE_{tra}	MSE_{tra}	% Improvement	MSE_{tra}	% Improvement
WM	9	0.00052	0.00042	19.2%	0.00046	11.5%
CH	9	0.00033	0.00041	−24.2%	0.00032	3.0%
D-Mogul	9	0.00035	0.00029	17.1%	0.00038	−8.5%

In view of these results, we can assert the following conclusion:

The granularity of an FRBS with good accuracy obtained with a specific learning method will not necessarily produce a FRBS with good behaviour if it is used with another learning method. The granularity of a specific problem depends not only on the problem itself but also on the learning method considered.

On the other hand, as shown in Table 4, the accuracy difference (both MSE_{tst} and MSE_{tra}) among the FRBSs obtained with different granularity levels justify the need for a granularity search to find the FRBS with best performance, or at least, with an appropriate one. Furthermore, this goal must be achieved in a reasonable amount of time. As we previously stated, it is difficult to run the learning method with all possible granularity levels, especially with non-deterministic methods. To solve this problem, an efficient granularity search method based on SA will be presented in the next section.

4. A SA based method to obtain a good uniform fuzzy partition granularity

Given an RB learning method and a specific problem, our goal is to find the optimal granularity level for each problem variable maintaining uniform fuzzy partitions. Therefore, each candidate solution is a concrete granularity level (the number of labels for each variable), and the cost function (Cost()) is based on the MSE_{tra} of the FRBS obtained with the RB learning method using that granularity.

This is an NP-problem, where considering seven possible values per variable ({3–9}) and N variables, the search space is composed of 7^N solutions. Therefore, we decided to tackle the problem by means of a heuristic search technique. Different possible choices are GAs [21], SA [1], Tabu search (TS) [13], among others. Since our goal is not only to obtain a good solution but also to obtain it quickly, we will consider a local search technique such as SA or TS, thus forgetting GAs for the sake of efficiency, even keeping in mind that they could be able to obtain very accurate solutions (GAs have been successfully applied to many problems in the FRBS design field [5,7,17,25,27]). In this paper, we will consider the SA procedure described in Appendix C.

The parameters of the SA procedure considered are shown in Table 11, with L being the number of possible values for the labels (seven in our case), with N being the number of variables of the problem considered, and with T_0, T_i, T_{i+1} being, respectively, the initial temperature, and the temperature in successive iterations. Finally, μ and ϕ are the parameters that influence the calculus of the initial temperature as described in Appendix C.

The number of iterations is calculated depending on the maximum number of solutions that can be generated and the number of solutions (state transitions) in each iteration. In our proposed SA procedure, there is a relaxation of the number of state transitions for each iteration. As described in Table 11, the number of accepted solutions is limited to N^2. We impose this constraint because, when the temperature is high at the beginning of the algorithm, a large number of accepted solutions could cause the procedure to move away from the optimal solution. To address this situation we must limit the number of accepted solutions per iteration. Other experiments were made changing the number of state transitions in each cooling (N^4), and considering other parameters in the initial temperature calculus ($\mu \in [0.1, 0.3]$ and $\phi \in [0.1, 0.3]$). However, the results are similar, and sometimes worse, with a higher run-time.

Table 11
Parameters of our SA procedure

Parameter	Value
Initial temperature	$T_0 = \frac{\mu}{-\ln(\phi)}\,\mathrm{Cost}(S_0)$ $\mu = 0.1,\ \phi = 0.1$ if $N > 3$ $\mu = 0.3,\ \phi = 0.3$ if $N \leqslant 3$
Decrement function of the temperature (cooling scheme)	$T_{i+1} = \alpha * T_i$ $\alpha = 0.9$
Maximum Number of state transitions in each iteration	N^3
Maximum number of acceptance solutions in each iteration	N^2
Maximum number of solutions that can be generated by the algorithm	L^N
Maximum number of iterations allowed without improvement	N

Table 12
Stopping criteria of our SA procedure

Number	Stopping criteria
1	The maximum number of iterations allowed without global improvement is reached
2	No solution was accepted in the last iteration
3	The maximum number of solutions have been generated

We considered three stopping criteria to reduce the run time of the procedure. They are shown in Table 12. It is interesting to note that in all the experiments performed, the procedure was always terminated by the first or second stopping criteria.

The implementation of our SA procedure incorporates a taboo record of explored solutions, along with their cost, in order to eliminate the possibility of redundant executions of the RB learning method, with the consequent saving of run time. We considered two possibilities for the initial solution: an information-based one, which considers the granularity with the same number of labels per variable producing the best $\mathrm{MSE}_{\mathrm{tra}}$ for the problem, and a random initial solution.

Four runs of the SA procedure have been made with different seeds for each RB learning method in the three problems considered. The solutions obtained by the SA are compared with the initial solution and the best solution found in Section 3.2 (see Table 4), the latter one denoted by S_{b}. Each table of results (Tables 16–23) has the following columns:

- The initial granularity (denoted by S_0).
- The granularity found by the SA procedure (denoted by S_{f}).
- The improvement percentage between the solution found and the initial solution (regarding to the $\mathrm{MSE}_{\mathrm{tra}}$).
- The worsening percentage between the solution found and the best granularity for this method (again regarding to the $\mathrm{MSE}_{\mathrm{tra}}$). This field does not appear in the D-Mogul method tables, due to the complexity of the best solution search, as previously said in Section 3.2.
- The number of solutions generated.
- The number of solutions evaluated (learning method runs).

The latter two fields allow us to know about the SA run time. The D-Mogul method table for the *rice taste evaluation* problem does not appear due to its

Table 13
Notation considered for the low voltage line length problem variables

Symbol	Meaning
x_1	Number of inhabitants of the town
x_2	Distance from the centre of the town to the three furthest clients
y	Total length of low voltage line installed

Table 14
Notation considered for the optimal electrical problem variables

Symbol	Meaning
x_1	Sum of the lengths of all streets in the town
x_2	Total area of the town
x_3	Area occupied by buildings
x_4	Energy supply to the town
y	Maintenance costs of medium voltage line

Table 15
Notation considered for the rice taste evaluation problem variables

Symbol	Meaning
x_1	Flavor
x_2	Appearance
x_3	Taste
x_4	Stickiness
x_5	Toughness
y	Global valorationp

huge run time. In every table, the four first lines correspond to the experiments of the first type of initialization, that which considers the same number of labels in each variable for the initial solution, and the next four lines are associated to the experiments considering a random initial solution.

According to the results of the SA procedure, we can state that our proposal appropriately satisfies the initial objective: "to find a good granularity level for a determinated problem and RB learning method in a reasonable time". The results are of significative importance in problems with high number of variables, where the number of possible granularity levels is high and the possibility of an exhaustive search is almost impossible. In the majority of the experiments developed for the *optimal electrical network* problem and the *rice taste evaluation* problem (5 and 6 variables, respectively), the optimal granularity was found with a low cost with respect to the size of the search space.

5. Concluding remarks

Our goal was to show the importance of the fuzzy partition granularity in the FRBS accuracy and to propose an efficient granularity search method. As a result, FRBSs with better performance can be designed without making any change to the RB learning method used, while maintaining uniform fuzzy partitions.

With regard to the influence of fuzzy partition granularity on the FRBSs behaviour, we can state that there is not an "absolute" granularity level that

Table 16
Results of the SA for the low voltage line length problem (WM)

Best solution (S_b): **6 9 9** – MSE_{tra}: **186904.3**
Column "% (1)": Improvement percentage: S_f vs S_0 (MSE_{tra})
Column "% (2)": Worsening percentage: S_f vs S_b (MSE_{tra})

Initial solution S_0			Final solution S_f			% (1)	% (2)	# Generated solutions	# Evaluated solutions
Granularity	MSE	Value	Granularity	MSE	Value				
9 9 9	tra	197613.4	6 6 9	tra	194271.9	1.6	3.9	83	43
	tst	283645.5		tst	149764.7				
9 9 9	tra	197613.4	6 9 9	tra	186904.3	5.4	0	84	49
	tst	283645.5		tst	264896.5				
9 9 9	tra	197613.4	8 9 9	tra	192980.7	2.3	3.2	44	18
	tst	283645.5		tst	230675.1				
9 9 9	tra	197613.4	8 9 9	tra	192980.7	2.3	3.2	59	32
	tst	283645.5		tst	230675.1				
8 7 7	tra	298553.6	6 6 9	tra	194271.9	53	3.9	109	54
	tst	303458.4		tst	149764.7				
3 5 3	tra	375882.5	8 9 9	tra	192980.7	94	3.2	107	67
	tst	466915.4		tst	230675.1				
3 8 6	tra	246841.8	6 6 9	tra	194271.9	27	3.9	136	77
	tst	216161.9		tst	149764.7				
9 4 4	tra	301927.0	6 7 5	tra	201031.3	50	7.5	202	64
	tst	292566.5		tst	254937.7				

Table 17
Results of the SA for the low voltage line length problem (CH)

Best solution (S_b): **8 8 6** – MSE_{tra}: **192498.2**
Column "% (1)": Improvement percentage: S_f vs S_0 (MSE_{tra})
Column "% (2)": Worsening percentage: S_f vs S_b (MSE_{tra})

Initial solution S_0			Final solution S_f			% (1)	% (2)	# Generated solutions	# Evaluated solutions
Granularity	MSE	Value	Granularity	MSE	Value				
8 8 8	tra	199421.3	8 8 6	tra	192498.2	3.5	0	123	60
	tst	180000.4		tst	167731.5				
8 8 8	tra	199421.3	8 8 8	tra	199421.3	0	3.5	42	21
	tst	180000.4		tst	180000.4				
8 8 8	tra	199421.3	8 8 6	tra	192498.2	3.5	0	61	34
	tst	180000.4		tst	167731.5				
8 8 8	tra	199421.3	8 8 6	tra	192498.2	3.5	0	50	33
	tst	180000.4		tst	167731.5				
8 7 7	tra	258462.3	8 8 6	tra	192498.2	34	0	41	21
	tst	258767.2		tst	167731.5				
3 5 3	tra	303417.1	5 7 8	tra	212979.2	42	10	97	58
	tst	328373.3		tst	190455.7				
3 8 6	tra	327818.1	5 7 8	tra	212979.2	35	10	77	41
	tst	307660.5		tst	190455.7				
9 4 4	tra	273605.4	9 6 5	tra	217234.1	25	12	79	48
	tst	292739.9		tst	208644.7				

Table 18
Results of the SA for the low voltage line length problem (D-Mogul)

Column "% (1)": Improvement percentage: S_f vs S_0 (MSE_{tra})

Initial solution S_0			Final solution S_f			% (1)	# Generated solutions	# Evaluated solutions
Granularity	MSE	Value	Granularity	MSE	Value			
9 9 9	tra	148068.6	8 8 9	tra	147250.9	0.5	60	24
	tst	205396.9		tst	179538.0			
9 9 9	tra	148068.6	8 8 9	tra	147250.9	0.5	42	22
	tst	205396.9		tst	179538.0			
9 9 9	tra	148068.6	8 8 9	tra	147250.9	0.5	42	20
	tst	205396.9		tst	179538.0			
9 9 9	tra	148068.6	9 9 9	tra	148068.6	0	30	20
	tst	205396.9		tst	205396.9			
8 7 7	tra	155275.0	8 7 7	tra	155275.0	0	37	20
	tst	187273.9		tst	187273.9			
3 5 3	tra	278654.9	4 7 4	tra	165021.5	40	64	40
	tst	341648.6		tst	212383.6			
3 8 6	tra	168056.4	8 9 8	tra	148066.6	11	93	50
	tst	152518.0		tst	195947.6			
9 4 4	tra	181030.7	7 7 6	tra	153500.9	15	63	42
	tst	200034.5		tst	206930.5			

Table 19
Results of the SA for the optimal electrical network problem (WM)

Best solution (S_b): **5 7 7 7 9** – MSE_{tra}: **24867.7**
Column "% (1)": Improvement percentage: S_f vs S_0 (MSE_{tra})
Column "% (2)": Worsening percentage: S_f vs S_b (MSE_{tra})

Initial solution S_0			Final solution S_f			% (1)	% (2)	# Generated solutions	# Evaluated solutions
Granularity	MSE	Value	Granularity	MSE	Value				
9 9 9 9 9	tra	32337.4	9 7 7 8 9	tra	26217.9	23	5.4	978	114
	tst	33504.9		tst	35800.8				
9 9 9 9 9	tra	32337.4	9 7 7 8 9	tra	26217.9	23	5.4	633	115
	tst	33504.9		tst	35800.8				
9 9 9 9 9	tra	32337.4	9 3 7 7 9	tra	25827.5	25	3.8	833	91
	tst	33504.9		tst	33205.3				
9 9 9 9 9	tra	32337.4	7 7 6 9 9	tra	26097.1	23	4.9	931	130
	tst	33504.9		tst	33403.5				
8 7 7 6 4	tra	112932.7	5 7 7 7 9	tra	24867.7	354	0	424	274
	tst	120243.1		tst	26964.1				
3 5 3 3 9	tra	127002.3	7 3 3 9 9	tra	25204.9	403	1.3	782	363
	tst	130291.1		tst	25915.1				
3 8 6 9 8	tra	31252.5	5 7 7 7 9	tra	24867.7	25	0	689	106
	tst	34787.8		tst	26964.1				
9 4 4 8 6	tra	73974.6	5 7 7 7 9	tra	24867.7	197	0	687	303
	tst	85733.3		tst	26964.1				

Table 20
Results of the SA for the optimal electrical network problem (CH)

Best solution (S_b): **5 6 9 9 7** – MSE_{tra}: **27698.0**
Column "% (1)": Improvement percentage: S_f vs S_0 (MSE_{tra})
Column "% (2)": Worsening percentage: S_f vs S_b (MSE_{tra})

Initial solution S_0			Final solution S_f			% (1)	% (2)	# Generated solutions	# Evaluated solutions
Granularity	MSE	Value	Granularity	MSE	Value				
8 8 8 8 8	tra	42735.8	5 6 9 9 7	tra	27698.0	54	0	1141	225
	tst	53596.9		tst	26134.3				
8 8 8 8 8	tra	42735.8	5 6 9 9 7	tra	27698.0	54	0	1187	228
	tst	53596.9		tst	26134.3				
8 8 8 8 8	tra	42735.8	5 6 9 9 7	tra	27698.0	54	0	1454	195
	tst	53596.9		tst	26134.3				
8 8 8 8 8	tra	42735.8	5 6 9 9 7	tra	27698.0	54	0	747	158
	tst	53596.9		tst	26134.3				
8 7 7 6 4	tra	151981.2	4 7 8 9 7	tra	27988.1	443	1.0	419	238
	tst	124062.0		tst	31105.6				
3 5 3 3 9	tra	138816.4	6 3 9 9 9	tra	27922.9	397	0.8	581	313
	tst	116088.5		tst	29592.0				
3 8 6 9 8	tra	55369.4	5 6 9 9 7	tra	27698.0	99	0	739	155
	tst	58860.5		tst	26134.3				
9 4 4 8 6	tra	77045.9	5 6 9 9 7	tra	27698.0	178	0	941	272
	tst	102226.2		tst	26134.3				

Table 21
Results of the SA for the optimal electrical network problem (D-Mogul)

Column "% (1)": Improvement percentage: S_f vs S_0 (MSE_{tra})

Initial solution S_0			Final solution S_f			% (1)	# Generated solutions	# Evaluated solutions
Granularity	MSE	Value	Granularity	MSE	Value			
8 8 8 8 8	tra	26690.7	3 3 7 9 9	tra	14523.9	44	1017	191
	tst	32783.0		tst	16810.1			
8 8 8 8 8	tra	26690.7	3 3 7 9 9	tra	14523.9	44	665	157
	tst	32783.0		tst	16810.1			
8 8 8 8 8	tra	26690.7	3 3 7 9 9	tra	14523.9	44	832	167
	tst	32783.0		tst	16810.1			
8 8 8 8 8	tra	26690.7	3 3 7 9 9	tra	14523.9	44	1243	234
	tst	32783.0		tst	16810.1			
8 7 7 6 4	tra	93259.3	3 3 7 9 9	tra	14523.9	84	1444	430
	tst	103968.3		tst	16810.1			
3 5 3 3 9	tra	77086.9	3 3 7 9 9	tra	14523.9	81	776	290
	tst	75432.9		tst	16810.1			
3 8 6 9 8	tra	24877.6	3 3 7 9 9	tra	14523.9	41	875	171
	tst	35779.6		tst	16810.1			
9 4 4 8 6	tra	58253.8	3 3 7 9 9	tra	14523.9	75	1232	267
	tst	82396.8		tst	16810.1			

Table 22
Results of the SA for the rice taste evaluation problem (WM)

Best solution (S_b): **9 9 9 8 7 9** – MSE_{tra}: **0.00042**
Column "% (1)": Improvement percentage: S_f vs S_0 (MSE_{tra})
Column "% (2)": Worsening percentage: S_f vs S_b (MSE_{tra})

Initial solution S_0			Final solution S_f			% (1)	% (2)	# Generated solutions	# Evaluated solutions
Granularity	MSE	Value	Granularity	MSE	Value				
9 9 9 9 9 9	tra	0.00052	7 8 9 8 9 9	tra	0.00043	17	2.3	1512	70
	tst	0.03403		tst	0.02816				
9 9 9 9 9 9	tra	0.00052	9 9 9 8 7 9	tra	0.00042	19	0	1080	50
	tst	0.03403		tst	0.03771				
9 9 9 9 9 9	tra	0.00052	7 8 9 8 9 9	tra	0.00043	17	2.3	1276	141
	tst	0.03403		tst	0.02816				
9 9 9 9 9 9	tra	0.00052	7 8 9 8 9 9	tra	0.00043	17	2.3	1512	70
	tst	0.03403		tst	0.02816				
8 7 7 6 4 6	tra	0.00168	9 9 9 8 7 9	tra	0.00042	75	0	1946	382
	tst	0.01097		tst	0.03771				
3 5 3 3 9 4	tra	0.00401	9 9 9 8 7 9	tra	0.00042	89	0	2614	961
	tst	0.00438		tst	0.03771				
3 8 6 9 8 8	tra	0.00150	7 8 9 8 9 9	tra	0.00043	71	2.3	1061	493
	tst	0.03159		tst	0.02816				
9 4 4 8 6 9	tra	0.00157	9 9 9 8 7 9	tra	0.00042	73	0	1209	403
	tst	0.01054		tst	0.03771				

Table 23
Results of the SA for the rice taste evaluation problem (CH)

Best solution (S_b): **9 9 9 9 7 9** – MSE_{tra}: **0.00032**
Column "% (1)": Improvement percentage: S_f vs S_0 (MSE_{tra})
Column "% (2)": Worsening percentage: S_f vs S_b (MSE_{tra})

Initial solution S_0			Final solution S_f			% (1)	% (2)	# Generated solutions	# Evaluated solutions
Granularity	MSE	Value	Granularity	MSE	Value				
9 9 9 9 9 9	tra	0.00033	9 9 9 9 7 9	tra	0.00032	3	0	1455	66
	tst	0.02029		tst	0.01084				
9 9 9 9 9 9	tra	0.00033	9 9 9 9 7 9	tra	0.00032	3	0	1439	80
	tst	0.02029		tst	0.01084				
9 9 9 9 9 9	tra	0.00033	9 9 9 9 7 9	tra	0.00032	3	0	1487	117
	tst	0.02029		tst	0.01084				
9 9 9 9 9 9	tra	0.00033	9 9 9 9 7 9	tra	0.00032	3	0	1496	63
	tst	0.02029		tst	0.01084				
8 7 7 6 4 6	tra	0.00102	9 9 9 9 7 9	tra	0.00032	68	0	973	313
	tst	0.01072		tst	0.01084				
3 5 3 3 9 4	tra	0.00353	9 9 9 9 8 9	tra	0.00034	90	6.2	938	676
	tst	0.00268		tst	0.01029				
3 8 6 9 8 8	tra	0.00085	9 9 9 9 7 9	tra	0.00032	62	0	2191	386
	tst	0.00090		tst	0.01084				
9 4 4 8 6 9	tra	0.00103	9 9 9 9 7 9	tra	0.00032	68	0	2590	466
	tst	0.00170		tst	0.01084				

generates the FRBS with best behaviour (both $\mathrm{MSE}_{\mathrm{tra}}$ and MSE_{test}) for all the learning methods. An appropriate DB depends not only on the problem, but also on the RB learning method considered. On the other hand, we have proved that the improvement obtained using a good fuzzy partition granularity is considerable. Therefore, we can assert that:

> The choice of the fuzzy partition granularity is an important task for the FRBS design, that should be considered since the beginning of the design process.

With respect to the granularity search method proposed, our SA procedure finds a good granularity level (in some cases the best) at a very low cost if compared with an exhaustive search. It is interesting to note that the best results were obtained in problems with a large number of variables, i.e., those presenting a greater search space. On the other hand, the parameters used in the SA procedure only depend on the number of variables of the problem considered.

The next step should be oriented to relax the form of the membership functions and to consider not only a different number of labels but also non-uniform fuzzy partitions. Our future work will be focused on this objective.

Appendix A. Learning methods

A.1. Wang and Mendel learning method (WM)

The ad hoc data covering RB generation process proposed by Wang and Mendel [31] have been widely known because of simplicity and good performance. The generation of the RB is put into effect by means of the following steps:

1. *Consider a fuzzy partition of the input variable spaces*: It may be obtained from the expert information (if it is available) or by a normalization process. If the latter is the case, perform a fuzzy partition of the input variable spaces dividing each universe of discourse into a number of equal or unequal partitions, select a kind of membership function and assign one fuzzy set to each subspace.
2. *Generate a preliminary linguistic rule set*: This set will be formed by the rule best covering each example (input–output data pair) contained in the input–output data set. The structure of these rules is obtained by taking a specific example, i.e., an $n+1$-dimensional real array (n input and 1 output values), and setting each one of the variables to the linguistic label best covering every array component.

3. *Give an importance degree to each rule*: Let $R_l = IF\, x_1$ *is* A_1 *and* ... *and* x_n *is* A_n *THEN* y *is* B be the linguistic rule generated from the example $e_l = (x_1^l, \ldots, x_n^l, y^l)$. The importance degree associated to it will be obtained as follows:

$$G(R_l) = \mu_{A_1}(x_1^l) \cdots \mu_{A_n}(x_n^l) \cdot \mu_{b}(y^l).$$

4. *Obtain a final RB from the preliminary fuzzy rule set*: The rule with the highest importance degree is chosen for each combination of antecedents.

A.2. Cordón and Herrera learning method (CH)

This method, proposed in [10], is an adaptation of the Ishibuchi's simplified TSK fuzzy rule generation method [18] that makes the process able to deal with rules with fuzzy consequent. It considers the n-dimensional table representation for the RB to generate and have two steps:

1. *Fill in the table*: The subset of the input–output data pairs belonging to the fuzzy input subspace associated to every cell of this table is considered.
2. *Choice of the rule consequent*: The consequent associated to the rule will be the output variable label that maximizes some covering criterion over the training set. No rules are generated in those cells where no data are located. Three possibilities for the covering criterion are presented next:
   - Maximum covering over the example set.
   - Maximum covering of the example best covered.
   - Average of the previous covering degrees.

In this paper, we have used the third one.

A.3. Descriptive-MOGUL learning method (D-Mogul)

The descriptive-MOGUL learning method [6] is based on the MOGUL paradigm presented in [9]. It allows us to automatically generate a complete KB when a training set is available. It consists of the following three steps:

1. An *iterative RB generation process* of desirable fuzzy rules able to include the complete knowledge of the example set.
2. A *genetic simplification process*, which finds the final RB able to approximate the input–output behaviour of the real system. It is based on eliminating some unnecessary rules from the rule set obtained in the previous stage, avoiding thus the possible over-fitting, by selecting the subset of rules best cooperating.
3. A *genetic tuning process* of the DB used that adjusts the membership functions in order to improve as far as possible the accuracy of the final KB.

In order to compare all the RBs obtained by considering uniform fuzzy partitions, the third step of this method has not been used in the experiments developed in this paper.

Appendix B. Problems used as benchmarks in this paper

B.1. Low voltage line length installed in a rural town

The first of the problems considered is that of finding a model that relates the total length of low voltage line installed in a rural town [8] with some characteristics of its (see Table 13). This model will be used to estimate the total length of line being maintained by an electrical company. We were provided with a sample of 495 towns in which the length of line was actually measured and the company used the model to extrapolate this length over more than 10,000 towns with these properties. We will limit ourselves to the estimation of the length of line in a town, given the inputs mentioned before. The training set contains 396 elements and the test set contains 99 elements.

B.2. Optimal electrical network for a town

The second problem has a different nature, since we will not deal with real data but with estimations of minimum maintenance costs which are based on a model of the optimal electrical network for a town [8]. These values are somewhat lower than the real ones, but companies are interested in an estimation of the minimum costs. Obviously, real maintenance costs are exactly accounted and hence a model that relates these costs to any characteristic of real towns would not be of great practical significance.

We were provided with data concerning four different characteristics of the towns and their minimum maintenance costs (see Table 14) in a sample of 1059 simulated towns. In this case, our objective was to relate the last variable (maintenance costs) with the other four ones. The training set contains 847 elements and the test set contains 212 elements.

B.3. Rice taste evaluation problem

The third problem deals with a subjective qualification of rice taste [18,23]. It is usually put into effect by means of the so-called *sensory test*. In this test, a group of experts, usually composed of 24 persons, evaluate the rice according to a set of characteristics associated to it (see Table 15). A sample with 105 evaluations of these experts is considered [23]. The training set contains 75 elements and the test set contains 30 elements.

Appendix C. Simulated annealing

SA [1] is derived from the analogy between statistical mechanics of particles of a substance (either liquid or solid) and the search for solutions in complex combinatorial optimization problems. Statistical mechanics addresses the behaviour of interacting particles of a substance. Different placements of particles in a substance yield different levels of energy. If the state of the substance is defined by the placement of its particles and thus its energy, the Metropolis algorithm is a mathematical model used to describe the transition of the substance from state i with energy $E(i)$ to state j with energy $E(j)$ at temperature T by a simple mechanism.

The Metropolis algorithm describes the process in which liquids crystallize: at high temperatures the energetic particles are free to move and rearrange; at low temperatures, the particles lose mobility as a result of decreasing energy, finally settling down to an equilibrium state resulting in the formation of a crystal having the minimum energy.

SA searches for the optimal solution or configuration of a combinatorial optimization problem. Let us suppose one needs to minimize a cost function described by many variables. A simple iterative scheme known as *local search* could be performed to find the minimum cost. During a local search process, an initial solution is given and then a new solution is proposed at random. If the cost of the new solution is lesser than that of the current solution, then the current solution is replaced by the new solution. If the cost of the new solution is higher than that of the current solution, a new solution is proposed again at random. This procedure continues until the solution with the minimum cost is found. Unfortunately, a *local search* may get stack at local minima. To alleviate the problem of getting trapped at local minima, SA occasionally allows "uphill moves" to solutions of higher cost. This is the essence of SA.

The acceptance probability of a new generated solution (S_{cand}), respect to the actual solution considered (S_{act}) is governed by the *Metropolis criterion*:

$$P_{\text{acc}}\,(S_{\text{cand}}) = \begin{cases} 1 & \text{if } \mathrm{Cost}(S_{\text{cand}}) < \mathrm{Cost}(S_{\text{act}}), \\ \exp(-\frac{\mathrm{Cost}(S_{\text{cand}})-\mathrm{Cost}(S_{\text{act}})}{T}) & \text{otherwise.} \end{cases}$$

SA requires two operations: a thermostatic operation known as a cooling schedule, which guides the decrease of the temperature, and a stocastic relaxation process that searches for the equilibrium solutions at each temperature. It can be demonstrated that SA is capable of reaching the optimal solutions asymptotically; that is, the proof assumes that the procedure undergoes an infinite number of transitions. To be practical, SA has to be implemented in finite time. Otherwise, it will not have any advantage over a very simple random search.

Therefore, it is necessary to specify an initial temperature, a cooling scheme to decrease the temperature, a criterion for determining the number of state transitions per temperature, the final temperature and the stopping criterion of the procedure. There are different cooling schedules proposed in the specialized literature [28]. As regards the initial temperature value, we will use the next formula:

$$T_0 = \frac{\mu}{-\ln(\phi)} \mathrm{Cost}(S_0)$$

with T_0 being the initial temperature, S_0 being the initial solution and ϕ being the probability of acceptance for a solution that can be μ per 1 worse than $\mathrm{Cost}(S_0)$. The latter two parameters are defined in the interval $[0,1]$.

The basic operation mode of SA, adapted to our problem, is described next:

```
INPUT(T_0, α, N, L)
T ← T_0
S_act ← Generate_Initial_Solution
solutions ← 1
S_best ← S_act
iterations_without_improv. ← 0
iteration_without_accepted_solution ← false
WHILE (solutions ⩽ L^N) AND (iterations_without_improv. < N)
AND NOT(iteration_without_accepted_solution) DO
   BEGIN
   best_improvement ← false
   accepted_solution_number ← 0
   count ← 0
   WHILE (count < N^3) AND (accepted_solution_number < N^2) DO
    BEGIN
    count ← count + 1
    S_cand ← Generate_Candidate_Solution(S_act)
    solutions ← solutions + 1
    δ ← cost(S_cand) − cost(S_act)
    IF (U(0,1) < e^(−δ/T)) OR (δ < 0)
       THEN BEGIN
          S_act ← S_cand
          accepted_solution_number ← accepted_solution_number + 1
          IF BETTER(S_act, S_best)
             THEN BEGIN
                S_best ← S_act
                best_improvement ← true
                END
          END
    END
   T ← α(T)
   IF (accepted_solution_number = 0)
    THEN iteration_without_accepted_solution ← true
   IF (best_improvement)
    THEN iterations_without_improv. ← 0
```

ELSE $iterations\_without\_improv. \leftarrow iterations\_without\_improv. + 1$
END
{Write as final solution, S_{best}}

References

[1] E.H.L. Aarts, Simulated Annealing and Boltzman Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing, Wiley, New York, Chicester, 1989.

[2] A. Bardossy, L. Duckstein, Fuzzy Rule-Based Modelling with Application to Geophysical, Biological and Engineering Systems, CRC Press, Boca Raton, 1995.

[3] P.P. Bonissone, P.S. Khedkar, Y.-T Chen, Genetic algorithms for automated tuning of fuzzy controllers, a transportation aplication, in: Proceedings of the Fifth IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'96, New Orleans, 1996, pp. 674–680.

[4] J. Casillas, O. Cordón, F. Herrera, A methodology to improve ad hoc data driven linguistic rule learning methods by inducing cooperation among rules, Technical Report #DECSAI-000101, University of Granada, Spain, February 2000.

[5] O. Cordón, F. Herrera, M. Lozano, On the combination of fuzzy logic and evolutionary computation: a short review and bibliography, in: W. Pedriycz (Ed.), Fuzzy Evolutionary Computation, Kluwer Academic Publishers, Dordrecht, 1997, pp. 55–77.

[6] O. Cordón, F. Herrera, A three-stage evolutionary process for learning descriptive and approximative fuzzy logic controller knowledge bases from examples, International Journal of Approximate Reasoning 17 (4) (1997) 369–407.

[7] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific, Singapore, 2000.

[8] O. Cordón, F. Herrera, L. Sánchez, Solving electrical distribution problems using hybrid evolutionary data analysis techniques, Applied Intelligence 10 (1999) 5–24.

[9] O. Cordón, M.J. del Jesus, F. Herrera, M. Lozano, MOGUL: a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning, International Journal of Intelligent Systems 14 (9) (1999) 1123–1153.

[10] O. Cordón, F. Herrera, A proposal for improving the accuracy of linguistic modeling, IEEE Transactions on Fuzzy Systems, 8 (3) (2000) 335–344.

[11] B. Filipic, D. Juricic, A genetic algorithm to support learning fuzzy control rules from examples, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzberg, 1996, pp. 403–418.

[12] P. Glorennec, Constrained optimization of FIS using an evolutionary method, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzberg, 1996, pp. 349–368.

[13] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, Dordrecht, 1997.

[14] R. Gudwin, F. Gomide, W. Pedrycz, Nonlinear context adaptation with genetic algorithms, International Journal of Intelligent Systems 13 (1998) 929–948.

[15] K. Hirota (Ed.), Industrial Applications of Fuzzy Technology, Springer, Berlin, Heidelberg, New York, 1993.

[16] F. Herrera, M. Lozano, J.L. Verdegay, Tuning fuzzy controllers by genetic algorithms, International Journal of Approximate Reasoning 12 (1995) 299–315.

[17] F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, 1996.

[18] H. Ishibuchi, K. Nozaki, H. Tanaka, Y. Hosaka, M. Matsuda, Empirical study on learning in fuzzy systems by rice test analysis, Fuzzy Sets and Systems 64 (1994) 129–144.

[19] H. Ishibuchi, T. Murata, A genetic-algorithm-based fuzzy partition method for pattern classification problems, in: F. Herrera, J.L. Verdegay (Eds.), Genetic Algorithms and Soft Computing, Physica-Verlag, Wurzberg, 1996, pp. 555–578.
[20] J.R. Jang, ANFIS: adaptive-network-based fuzzy inference system, IEEE Transactions on Systems, Man, and Cybernetics 23 (3) (1993) 665–684.
[21] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer, Berlin, Heidelberg, New York, 1996.
[22] H. Nomura, L. Hayashi, N. Wakami, A learning method of fuzzy inference rules by descent method, in: Proceedings of the First IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'92, San Diego, 1992, pp. 203–210.
[23] K. Nozaki, H. Ishibuchi, H. Tanaka, A simple but powerful heuristic method for generating fuzzy rules from numerical data, Fuzzy Sets and Systems 86 (1997) 251–270.
[24] W. Pedrycz (Ed.), Fuzzy Modelling. Paradigms and Practice, Kluwer Academic Publishers, Dordrecht, 1996.
[25] W. Pedrycz (Ed.), Fuzzy Evolutionary Computation, Kluwer Academic Publishers, Dordrecht, 1997.
[26] W. Pedrycz, R. Gudwin, F. Gomide, Nonlinear context adaptation in the calibration of fuzzy sets, Fuzzy Sets and Systems 88 (1997) 91–97.
[27] E. Sánchez, T. Shibata, L.A. Zadeh (Eds.), Genetic Algorithms and Fuzzy Logic Systems. Soft Computing Perspectives, World Scientific, Singapore, 1997.
[28] G. Sorkin, Efficient simulated annealing on fractal energy landscapes, Algorithmica 6 (1991) 367–418.
[29] H. Takagi, N. Suzuki, T. Koda, Y. Kojima, Neural networks designed on approximate reasoning architecture and their applications, IEEE Transactions on Neural Networks 3 (5) (1992) 752–760.
[30] L.J.R. Velasco, S. López, L. Magdalena, Genetic fuzzy clustering for the definition of fuzzy sets, in: Proceedings of the Sixth IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'97, Barcelona, 1997, pp. 1665–1670.
[31] L.X. Wang, J.M. Mendel, Generating fuzzy rules by learning from examples, IEEE Transactions on Systems, Man, and Cybernetics 22 (1992) 1414–1427.
[32] Y. Yoshinari, W. Pedrycz, K. Hirota, Construction of fuzzy models through clustering techniques, Fuzzy Sets and Systems 54 (1993) 157–165.
[33] L. Zheng, A practical guide to tune proportional and integral (PI) like fuzzy controllers, in: Proceedings of the First IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'92, San Diego, 1992, pp. 633–640.